UCP-Compliant MCP E-Commerce Platform – Implementation

Overview

This document describes a **phased, production-grade implementation plan** for building a **UCP-compliant MCP server** to power a secure, scalable, and extensible **AI-driven e-commerce chatbot**.

The plan is designed for: - **Zero-trust security by default** - **Strict UCP protocol compliance** - **Operational reliability** (idempotency, webhooks, retries, cleanup)

Architecture: **Java (Spring Boot MCP Tool Server) + Node/NestJS Orchestrator + Prisma + Redis**

Security Hardening

Priority: CRITICAL

0.1 Add Spring Security to MCP Server

Goal: Enforce authenticated access to all MCP tool executions.

Changes - Add Spring Security dependency - Implement JWT authentication filter - Whitelist only health & discovery endpoints

Files - `pom.xml` - `SecurityConfig.java` - `orchestrator.service.ts`

Rules - Public: - `/actuator/health` - `/.well-known/ucp` - Secured: - `/tools/execute/**`

Orchestrator Update - Attach `Authorization: Bearer <token>` to all MCP tool calls

Lock Down CORS

Goal: Prevent cross-origin abuse.

Change - Replace `allowedOrigins("*")` with env-driven whitelist

Files - `AppConfig.java`

Env Var

```
ALLOWED_ORIGINS=http://localhost:3001,http://localhost:3000
```

0.3 Rate Limiting

Goal: Protect MCP tools from abuse and LLM loops.

Implementation - Library: Bucket4j - Algorithm: Token Bucket - Storage: Redis

Limits - 100 req/min per **user** - 200 req/min per **IP**

Files - `RateLimitFilter.java`

UCP Discovery Endpoint

Priority: HIGH

1.1 Database Schema Updates

Goal: Store UCP metadata per provider.

Schema (`schema.prisma`)

```
ucp_profile  Json?   @default("{}")
auth_type    String? @default("none")
oauth_config Json?   @default("{}")
```

Migration

```
cd apps/bff-node
npx prisma migrate dev --name add_ucp_metadata
```

1.2 Discovery Controller

Endpoint

```
GET /.well-known/ucp
```

Responsibilities - Advertise platform capabilities - Expose registered tools - Declare authentication requirements

Response Includes - Platform metadata - Supported capabilities - Tool schemas - Enabled providers - UCP version metadata

Caching - Redis TTL: **5 minutes**

Files - `DiscoveryController.java`

Phase 2 – UCP Checkout Session Pattern

Priority: HIGH
Breaking: Yes (Feature-flagged)

2.1 Checkout Session Database Model

Purpose: Freeze cart state and pricing during checkout.

Model: `checkout_sessions` - Explicit lifecycle: CREATED → COMPLETED / CANCELLED / EXPIRED - 30-minute expiry - Full pricing snapshot

Indexes - `user_id` - `status` - `expires_at`

2.2 MCP Checkout Tools

Service: `CheckoutService.java`

Tools

1. `commerce.checkout.create`
2. `commerce.checkout.update`
3. `commerce.checkout.get`
4. `commerce.checkout.complete`
5. `commerce.checkout.cancel`

Key Rules - Prices frozen at creation - Expiration extended on updates - Idempotency enforced on completion - Orders > ₹50,000 require confirmation

2.3 Human-in-the-Loop Confirmation

Trigger - `total.amount > 50,000`

Flow - Orchestrator pauses tool execution - UI prompts user confirmation - Timeout: 2 minutes

Files - `orchestrator.service.ts` - `ChatApp.tsx`

2.5 Product Discovery Extensions

New Tools

- `commerce.product.estimateShipping`

- `commerce.product.listVariants`

- `commerce.promotions.get`

- `commerce.promotions.validateCoupon`

New Table: `coupons`

Phase 3 – OAuth2 Identity Linking

Priority: MEDIUM
Breaking: No

3.1 OAuth Token Storage

Table: `provider_auth_tokens`

Security - AES-256 encrypted tokens - Auto-refresh before expiry

3.2 OAuth2 Service

Endpoints - `/v1/oauth/:providerId/authorize` - `/v1/oauth/:providerId/callback`

Implementation - passport-oauth2 - CSRF state in Redis (10 min TTL)

3.3 Admin UI

Features - Connect / Disconnect provider - Token expiry display - Connection status

File - `Providers.tsx`

Phase 4 – Operational Excellence (Week 6)

4.1 Idempotency Enforcement

Service: `IdempotencyService.java`

Applies To - Checkout completion - Order creation

TTL: 24 hours

4.2 Webhook Infrastructure

Tables - `webhook_events` - `webhook_subscriptions`

Features - HMAC-SHA256 signatures - Exponential retries - Event persistence

4.3 Background Cleanup Jobs

Schedules - Checkout expiry: every 15 min - Idempotency cleanup: daily 2 AM - Webhook cleanup: daily 3 AM

5.2 Documentation

New Docs - `UCP_COMPLIANCE.md` - `MIGRATION_GUIDE.md` - `WEBHOOKS.md`

Environment Variables Summary

```
JWT_SECRET=
ALLOWED_ORIGINS=
PROVIDER_RD_AUTH_TOKEN=
TOKEN_ENCRYPTION_KEY=
OAUTH_CLIENT_ID=
OAUTH_CLIENT_SECRET=
ENABLE_CHECKOUT_FLOW=true
ENABLE_OAUTH_PROVIDERS=false
ENABLE_WEBHOOKS=false
IDEMPOTENCY_ENFORCEMENT=true
```